



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/705,754	11/10/2003	Robert E. Viehland	MS1-1746US	4269
22801	7590	10/10/2007		
LEE & HAYES PLLC 421 W RIVERSIDE AVENUE SUITE 500 SPOKANE, WA 99201			EXAMINER YAARY, MICHAEL D	
			ART UNIT 2193	PAPER NUMBER
			MAIL DATE 10/10/2007	DELIVERY MODE PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary

Application No.

10/705,754

Applicant(s)

VIEHLAND ET AL.

Examiner

Michael Yaary

Art Unit

2193

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 13 August 2007.
- 2a) ☐ This action is FINAL. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-5,8-10,12,13,15-26 and 28-38 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-5,8-10,12,13,15-26 and 28-38 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
 - ☐ Certified copies of the priority documents have been received in Application No. _____.
 - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- ☐ Notice of References Cited (PTO-892)
- ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- ☐ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____
- ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____
- ☐ Notice of Informal Patent Application
- ☐ Other: _____

DETAILED ACTION

1. Claims 1-5, 8-10, 12, 13, 15-26, and 28-38 are pending in the application.

Claim Rejections - 35 USC § 101

2. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

3. As to claims 12-19, the claims are rejected under U.S.C. 101 as the claimed "device" is software per se, as it is not tangibly embodied, failing to recite any hardware as part of the device. The supporting hardware features of the device, as disclosed in the specification, need to be brought into the claims in order to overcome the 35 U.S.C. 101 rejection.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

Art Unit: 2193

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

5. Claims 1-5 and 8 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bates et al. (hereafter Bates(I))(US Pat. 6,077,312) in view of Bates et al. (hereafter Bates(II))(US Pat. 6,493,834).

6. Bates(I) and Bates(II) were cited in the previous office action dated 03/13/2007.

7. **As to claim 1**, Bates(I) discloses one or more processor-accessible storage media comprising processor executable instructions that, when executed, direct a device to perform actions (abstract, lines 1-5), comprising:

Determining if an instruction of a line of common intermediate language code meets predetermined related criterion (Column 2, lines 4-12 and lines 40-45 disclose in object-oriented type of programming, which resembles the type of code of common intermediate language, determining if instructions meet a predetermined criteria.); and

If so injecting a decision point in association with the instruction of the line of common intermediate language code (Column 3, lines 10-18 disclose utilizing breakpoints (decision points) with the instructions that meet predetermined criteria.); and an instruction type indicator that indicates an instruction type for the line of common intermediate language code (Column 6, lines 3-23 disclose instructions being analyzed to determine a routine and specific instruction, thus implicitly determining the type of

instruction, and placing an indicator would have been obvious to one of ordinary skill in the art in order to identify the specific routines and instructions thereof.).

8. Bates(I) does not disclose that the predetermined criterion is exception-related criterion; the decision point enables a decision as to whether an exception is to be thrown with respect to the instruction; and wherein the action of injecting a decision point comprises an action of injecting a bookmark entry.

However, Bates(II) discloses that the predetermined criterion is exception-related criterion (Column 4, lines 19-22 disclose exception related instructions); the decision point enables a decision as to whether an exception is to be thrown with respect to the instruction (column 6, lines 57-59); and wherein the action of injecting a decision point comprises an action of injecting a bookmark entry (Column 7, lines 1-3 disclose a breakpoint manager, thus being analogous to the bookmark entry that discloses details of the instruction.).

9. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the teachings of Bates(I), by utilizing exception related instructions and a bookmark entry, as taught by Bates(II), for the benefit of creating more resourceful methods and tools used in the debugging and testing of computer programs (Bates(II), column 2, lines 14-17).

Art Unit: 2193

10. **As to claim 2**, Bates(II) discloses retrieving the line of common intermediate language code from a common intermediate language code program prior to the determining (Step 920, fetch next instruction, of figure 9 discloses retrieving the instruction before interpreting (determining).); retrieving another line of common intermediate language code from the common intermediate language code program (Step 910 of figure 9 discloses checking for more lines of instructions, if so fetching or retrieving them.); and repeating the action of determining and the action of injecting a decision point for an instruction of the retrieved other line of common intermediate language code (column 9, line 67-column 10, line 10).

11. **As to claim 3**, Bates(II) discloses determining if the instruction of the line of common intermediate language code is capable of throwing an exception (column 10, lines 5-10).

12. **As to claim 4**, Bates(II) discloses determining if the instruction of the line of common intermediate language code is capable of throwing an exception (column 10, lines 5-10) and is related to a pre-selected area (column 8, lines 58-67).

13. **As to claim 5**, Bates(II) discloses determining if the instruction of the line of common language code is capable of throwing an exception with reference to a common intermediate language code specification (column 8, lines 58-61 and column 9, lines 12-14).

14. **As to claim 8**, Bates(II) discloses an identifier that uniquely identifies the decision point within the common intermediate language code that is being instrumented (column 11, lines 43-48).

15. Claims 10 and 28 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bates(I) in view of Bates(II) as applied to claim 1 above, and further in view of Angel et al. (hereafter Angel)(US Pat. 6,314,558).

16. Angel was cited in the previous office action dated 03/13/2007.

17. **As to claim 10**, The combination of Bates(I) and Bates(II) disclose repeating the actions of determining and injecting a decision point for a plurality of respective instructions of a plurality of respective lines of the code (Bates(II) column 9, line 64-column 10, line 10).

18. The combination of Bates(I) and Bates(II) do not disclose an instrumentation tool that produces instrumented common intermediate language code from common intermediate language code (column 3, lines 16-20).

However, Angel discloses an instrumentation tool that produces instrumented common intermediate language code from common intermediate language code (column 3, lines 16-20).

19. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the teachings of Bates(I) and Bates(II), by producing common instrumented intermediate language code, as taught by Angel, in order to facilitate debugging.

20. **As to claim 28**, the combination of Bates(I) and Bates (II) disclose one-or more processor-accessible storage media comprising an instrumentation tool that is capable of determining whether respective instructions from common intermediate language code meet at least one predetermined criterion (Bates(I) column 2, lines 4-12 and lines 40-45 disclose in object-oriented type of programming which resembles the type of code of common intermediate language, determining if instructions meet a predetermined criteria.) and that is adapted to inject respective decision points into the common intermediate language code in association with the respective instructions that meet the at least one predetermined related criterion (Bates(I) column 3, lines 10-18 disclose utilizing breakpoints (decision points) with the instructions that meet predetermined criteria.); and that the predetermined related criterion is predetermined exception-related criterion (Bates(II) column 4, lines 19-22 disclose exception related instructions), each injected respective decision point including an identifier of the injected respective

Art Unit: 2193

decision point (Bates (II) column 11, lines 43-48) and a call to a program that can selectively cause an exception to be thrown with respect to the respective associated instruction (Bates(II) column 9, line 64-column 10, line 10.); and each injected respective decision point including an indication of an instruction type of the respective associated instruction (Column 6, lines 3-23 disclose instructions being analyzed to determine a routine and specific instruction, thus implicitly determining the type of instruction, and placing an indicator would have been obvious to one of ordinary skill in the art in order to identify the specific routines and instructions thereof.).

21. Claims 12, 13, and 15-26, 29-31, 33, 34, and 36-38 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bates(II) in view of Angel and Bates(I).

22. **As to claim 12**, Bates(II) discloses code that includes a test couplet corresponding to a decision point and an associated instruction (Inherent in column 6, lines 57-59 and column 10, lines 14-36 as the decision point (breakpoint) always corresponds to an instruction, thus forming a couplet.) and to evaluate the test couplet to selectively decide whether to throw an exception with respect to the associated instruction (column 9, lines 12-14 and column 10, lines 5-10.); a bookmark entry (Column 7, lines 1-3 disclose a breakpoint manager, thus being analogous to the bookmark entry that discloses details of the instruction.) and an identifier of the decision point (column 11, lines 43-48).

23. Bates(II) does not disclose that the code is an instrumented common intermediate language code; a decision runtime library that is adapted to evaluate the test couplet; a common language runtime component that interprets the decision point so as to call the decision runtime library prior to executing the associated instruction.

However, Angel discloses that the code is an instrumented common intermediate language code (column 3, lines 16-20); a decision runtime library that is adapted to evaluate the test couplet (column 18, lines 16-19); and a common language runtime component that interprets the decision point so as to call the decision runtime library prior to executing the associated instruction (column 18, lines 27-33).

24. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the teachings of Bates(II), by incorporating a runtime library, as taught by Angel, for the benefit of being able to perform the evaluating and instrumentation methods during runtime.

25. The combination of Bates(II) and Angel do not disclose the bookmark entry comprises an indication of an instruction type of the associated instruction.

However, Bates(I) discloses the bookmark entry comprises an indication of an instruction type of the associated instruction (Column 6, lines 3-23 disclose instructions being analyzed to determine a routine and specific instruction, thus implicitly determining the type of instruction, and placing an indicator would have been obvious to

Art Unit: 2193

one of ordinary skill in the art in order to identify the specific routines and instructions thereof.).

26. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the teachings of Bates(II) and Angel, by determining the instruction type, as taught by Bates(I), in order to be able to specifically identify routines and instructions to be used.

27. **As to claim 13**, Bates(II) further discloses the instrumented common language code is in a binary form (column 19, lines 38-41).

28. **As to claim 15**, Bates(II) discloses to evaluate the test couplet to selectively decide whether to throw an exception responsive to the bookmark entry (column 6, lines 57-59 and column 10, lines 5-10), and Angel discloses the decision runtime library is further adapted to evaluate the test couplet (column 18, lines 10-33).

29. **As to claim 16**, it is rejected in the same manner as claim 15 above, in addition, Bates(II) further discloses whether to throw an exception based on throw exception logic (column 10, lines 37-55).

30. **As to claim 17**, Bates(II) further discloses to throw an exception based on at least one throw exception decision logic factor selected from the group comprising:

Art Unit: 2193

Throwing an exception when first encountering a given decision point using an identifier of the given decision point (column 10, lines 37-55).

31. **As to claim 18**, Bates(II) discloses to evaluate the test couplet to selectively decide whether to throw an exception (column 6, lines 57-59 and column 10, lines 5-10), and Angel discloses decision runtime library is further adapted to evaluate the test couplet (column 18, lines 10-33) and that it responsive to the indication of the instruction type (column 9, lines 28-32).

32. **As to claim 19**, Bates(II) discloses modularizing by exception category (column 8, lines 60-62), and Angel discloses a modularized decision runtime library (column 18, lines 16-19).

33. **As to claim 20**, the combination of Bates(II), Angel, and Bates(I) disclose an arrangement for enabling reliability testing of managed code (Bates(II), column 5, line 65-column 6, line 8), the arrangement comprising:

A plurality of decision points and decision means for deciding whether to throw an exception at each decision point of the plurality of decision points (Bates(II), column 9, line 64-column 10, line 36), instrumenting means for instrumenting common intermediate language code to produce instrumented common intermediate language code (Angel, column 3, lines 16-20); and analysis means for analyzing whether individual instructions of a plurality of instructions can result in a failure (Bates(II),

Art Unit: 2193

column 9, line 67-column 10, line 10) and injection means for injecting a respective decision point in association with each respective individual instruction (Bates(II), column 10, lines 14-36), and the plurality of instructions is of common intermediate language code (Angel, column 3, lines 16-20); and means for injecting a respective bookmark entry that indicates an instruction type of the respective individual instruction associated with respective decision point (Bates(I) column 6, lines 3-23 disclose instructions being analyzed to determine a routine and specific instruction, thus implicitly determining the type of instruction, and placing an indicator would have been obvious to one of ordinary skill in the art in order to identify the specific routines and instructions thereof.) and a plurality of decision points and decision means for deciding whether to throw an exception at each decision point of the plurality of decision points (Bates(II), column 9, line 64-column 10, line 36).

34. **As to claims 21 and 22**, Bates(II) discloses means the respective bookmark entry further identifies the respective decision point and means for injecting a call at least one module that is capable of evaluating the respective decision point with regard to whether a failure is to be induced (column 10, lines 14-36 disclose the induced failure due to the decision point, or thrown exception due to breakpoint).

35. **As to claim 23**, Angel discloses common language runtime means for executing the instrumented common intermediate language code and the decision means in a runtime environment (abstract, lines 1-19).

36. **As to claim 24**, Bates(II) discloses the decision means at each decision point of the plurality of decision points being called (Column 9, line 64-column 10, line 10 disclose making a decision, by interpreting each instruction, if an exception throw is possible and thus inserting breakpoints (decision points).), and Angel discloses that the common language runtime means calls the decision means (column 3, line 66-column 4, line 8).

37. **As to claim 25**, it is rejected in the same manner as claim 16 above.

38. **As to claim 26**, Bates(II) further discloses the arrangement comprises at least one device having processor-accessible storage media (column 5, line 65-column 6, line 8 and column 6, lines 34-45).

39. **As to claim 29**, the combination of Bates(II), Angel, and Bates(I) disclose determining whether an instruction is capable of causing an exception (Bates(II), column 10, lines 3-10); and if so, injecting a decision point in association with the instruction to mark the instruction for evaluation during execution (Bates(II), column 10, lines 8-10 disclose implementing a breakpoint (decision point) if the instruction can throw an exception), the evaluation involving a decision as to whether a failure is to be induced with respect to the instruction (column 10, lines 14-36 disclose the induced failure due to the decision point, or thrown exception due to breakpoint), and the

instruction is from common intermediate language code and the evaluation is during a common language runtime execution (Angel, Column 3, lines 16-20 discloses selecting portions of byte code to be instrumented. Common intermediate language code is a form of byte code); and the injecting a decision point comprises injecting an indicator of an instruction type of the instruction (Bates(I) column 6, lines 3-23 disclose instructions being analyzed to determine a routine and specific instruction, thus implicitly determining the type of instruction, and placing an indicator would have been obvious to one of ordinary skill in the art in order to identify the specific routines and instructions thereof.)

40. **As to claim 30**, Bates(II) further discloses determining whether the instruction from the common intermediate language code is capable of causing an exception (column 10, lines 5-10) and is related to a pre-selected exception category (column 8, lines 58-62).

41. **As to claim 31**, Bates(II) discloses injecting a identifier of the decision point (column 11, lines 43-48).

42. **As to claim 33**, Bates(II) further discloses one or more processor-accessible storage media comprising processor-executable instructions that, when executed, direct a device to perform the method recited in claim 29 (column 6, lines 34-45).

43. **As to claim 34**, Bates(II) discloses repeating the determining and injecting a decision point for a plurality of instructions (column 9, line 67-column 10, line 5), and Angel discloses that the plurality of instructions are from the common intermediate language code (column 3, lines 16-18 disclose instrumenting byte code) and producing instrumented common language code as a result of the repeating (column 3, lines 18-20).

44. **As to claim 36**, Bates(II) further discloses selectively deciding whether the execution is to fail at the decision point (column 10, lines 14-36 disclose the induced failure due to the decision point, or thrown exception due to breakpoint).

45. **As to claim 37**, Bates(II) further discloses if it is decided at the selectively deciding that the execution is to fail at the decision point, then choosing which exception of at least two exceptions is to be thrown (column 9, lines 12-14).

46. **As to claim 38**, Bates(II) further discloses if it is decided at the selectively deciding that the execution is to fail at the decision point, then including a failure in the execution of the common language runtime with respect to the instruction (column 10, lines 14-36).

47. Claim 9 is rejected under 35 U.S.C. 103(a) as being unpatentable over Bates(I) in view of Bates(II), applied to claim 1 above and further in view of Inamdar (US Pub. 2003/0149960).

48. Inamdar was cited in the previous office action dated 03/13/2007.

49. **As to claim 9**, Bates(I) and Bates(II) disclose a program that is adapted to evaluate whether the exception is to be thrown with respect to the instruction (Bates(II) column 9, line 67-column 10, line 10).

50. Bates(I) and Bates(II) do not disclose injecting a call to a decision runtime library and the decision runtime library comprising the program that does the evaluating.

However, Inamdar discloses injecting a call to a decision runtime library and the decision runtime library comprising the program that does the evaluating ([0022], lines 1-18 and [0040], lines 6-8 disclose instrumenting code into instructions based on a probe and runtime library, thus reading on injecting a call to a runtime library in order to evaluate instructions).

51. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the teachings of Bates(I) and Bates(II), by implementing a runtime library used in code instrumentation, as taught by Inamdar, for the benefit of performing the evaluating and instrumentation methods during runtime.

52. Claims 32 and 35 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bates(II), Angel, and Bates(I), as applied to claim 29 above, and further in view of Inamdar.

53. **As to claim 32**, the combination of Bates(II), Angel, and Bates(I) do not disclose injecting a call to at least one module of a decision runtime library that is adapted to perform the evaluation.

However, Inamdar discloses injecting a call to at least one module of a decision runtime library that is adapted to perform the evaluation ([0022], lines 14-17).

54. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the teachings of Bates(II), Angel, and Bates(I), by injecting a call to a module of a runtime library, as taught by Inamdar, for the benefit of performing the evaluating and instrumentation methods during runtime.

55. **As to claim 35**, the combination of Bates(II), Angel, Bates(I), and Inamdar disclose detecting the decision point (Bates(II), column 11, lines 43-48) in the instrumented common intermediate language (Angel, column 3, lines 16-20) during the execution thereof, and calling at least one module of a decision runtime library, which is

adapted to perform the evaluation, as a result of the detecting (Inamdar, [0022], lines 10-17).

Response to Arguments

56. Applicant's arguments with respect to claims 1-5, 8-10, 12, 13, 15-26, and 28-38 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael Yaary whose telephone number is (571) 270-1249. The examiner can normally be reached on Monday-Friday, 8:00 a.m - 5:00 p.m..

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on (571) 272-3756. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Art Unit: 2193

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

MY
MY


MENG-AL T. AN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100